horizontal line

**Assignment 07 | MFP**

**CE-092**

Assignment submission for Microprocessor Fundamentals and Programming subject week 7.

nevilparmar24@gmail.com

**─**

**Task 1:**

Write a program to check whether a given number is an Armstrong number.Display appropriate message.

Code:

data segment

num *db* "371"

str1 *db* "Number is not armstrong$"

str2 *db* "Number is armstrong$"

data *ends*

code segment

assume cs:code,ds:data

start:

mov ax,data

mov ds,ax

mov si,offset num

mov ax,0000h

mov cx,0003h

next:

mov ax,0000h

mov bl,[si]

and bl,0fh

mov al,bl

mul bl

mul bl

add dx,ax

inc si

Loop next

cmp dx,371d

jnz notarm

mov dx,offset str2

jmp ex

notarm:

mov dx,offset str1

jmp ex

ex:

mov ah,9h

int 21h

mov ax,4c00h

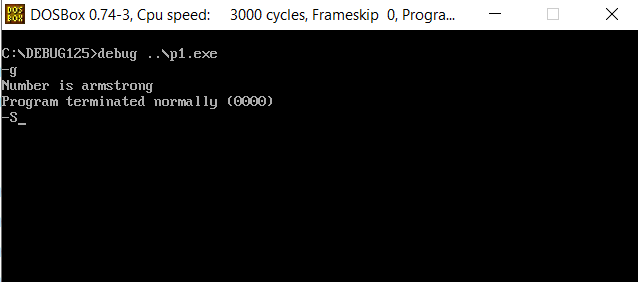
int 21h

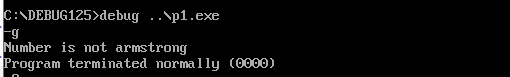
code *ends*

end start

Output:

**Check for Num = 371**

**Check for Num = 105**

****

**Task 2:**

Write a program to check whether two strings are the same irrespective of their case.

Code:

data segment

str1 *db* "abc"

len1 *db* $-str1

str2 *db* "ABd"

len2 *db* $-str2

msg1 *db* "Equal",13,10,"$"

msg2 *db* "Not Equal" ,13,10,"$"

lookUp *db* "abcdefghijklmnopqrstuvwxyz"

data *ends*

code segment

assume cs:code,ds:data

mov ax,data

mov ds,ax

mov al,len1

cmp al,len2

jnz notequal

mov bx,offset lookUp

mov si,offset str1

mov di,offset str2

mov cl,len1

start:

mov al,[si]

cmp al,'a'

jl S1

CONTINUE1:

mov dl,al

mov al,[di]

cmp al,'a'

jl S2

CONTINUE2:

cmp al,dl

jnz notequal

inc si

inc di

loop start

jmp equal

notequal:

mov dx,offset msg2

mov ah,09h

int 21h

jmp stop

equal:

mov dx,offset msg1

mov ah,09h

int 21h

stop:

mov ax,4c00h

int 21h

S1:

sub al,'A'

xlat

jmp CONTINUE1

S2:

sub al,'A'

xlat

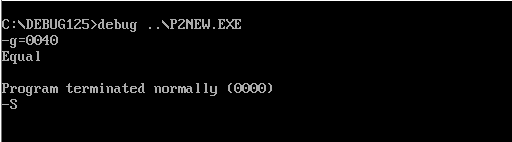
jmp CONTINUE2

code *ends*

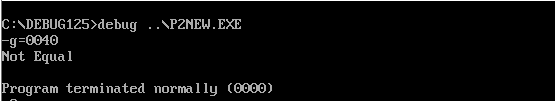
end

Output:

**Str1 = “abc” & str2 = “ABC”**



**Str1 = “abc” & str2 = “ABd”**

****

**Task 3:**

Write a Program to check whether the input string is palindrome or not. Get the string through the user.

Code:

DATA SEGMENT

BLOCK1 *DB* 'MALAYALAM'

MSG1 *DB* "It is palindrome $"

MSG2 *DB* "It is not palindrome $"

DATA *ENDS*

PRINT MACRO MSG

MOV AH,09H

LEA DX,MSG

INT 21H

INT 3H

ENDM

EXTRA SEGMENT

BLOCK2 *DB* 9 DUP(?)

EXTRA *ENDS*

CODE SEGMENT

ASSUME CS:CODE,DS:DATA,ES:EXTRA

START:

MOV AX,DATA

MOV DS,AX

MOV AX,EXTRA

MOV ES,AX

LEA SI,BLOCK1

LEA DI,BLOCK2+8

MOV CX,00009H

BACK:

*CLD*

LODSB

*STD*

STOSB

LOOP BACK

LEA SI,BLOCK1

LEA DI,BLOCK2

MOV CX,0009H

*CLD*

REPZ CMPSB

JNZ SKIP

PRINT MSG1

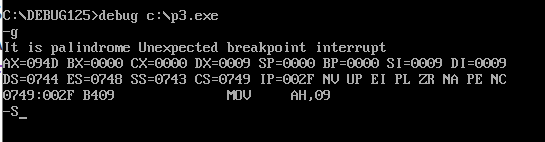
SKIP:

PRINT MSG2

CODE *ENDS*

END START

Output:



**Task 4:**

Write a program to search an element from a sorted list using binary search.

Code:

data segment

array *dw* 0000h,0001h,0002h,0003h,0004h,0005h,0006h,0007h,0008h,0009h

len *dw* ($-array)/2

key equ 0007h

msg *db* "key is found at "

res *db* " position",13,10," $"

notfoundstr *db* 'not found $'

data *ends*

code segment

assume ds:data,cs:code

start:

mov ax,data

mov ds,ax

mov bx,00

mov dx,len

mov cx,key

binary\_search:

cmp bx,dx

ja notfound

mov ax,dx

add ax,bx

shr ax,1

mov si,ax

add si,si

cmp cx,array[si]

jae big

dec ax

mov dx,ax

jmp binary\_search

big:

je success

inc ax

mov bx,ax

jmp binary\_search

success:

add al,01

add al,'0'

mov res,al

lea dx,msg

jmp result

notfound:

lea dx,notfoundstr

result:

mov ah,09h

int 21h

mov ah,4ch

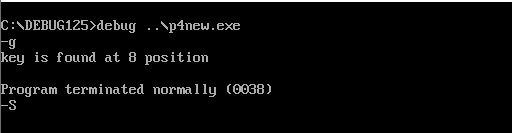
int 21h

code *ends*

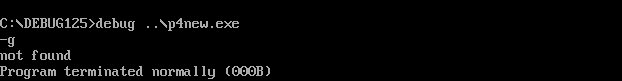
end start

Output:

**Search for key = 0007h**



**Search for key = 0010h**



Nevil Parmar

CE-092

https://nevilparmar.me